The ACE 2005 (ACE05) Evaluation Plan

Evaluation of the Detection and Recognition of ACE ***Entities, Values***, ***Temporal Expressions***, ***Relations,*** and ***Events***

1 INTRODUCTION

The objective of the ACE program is to develop automatic content extraction technology to support the automatic processing of source language data. Possible down-stream processing includes classification, filtering, and selection based on the content of the source data, i.e., based on the meaning conveyed by the language. Thus, the ACE program is dedicated to the development of technologies that automatically infer meaning from language data.

2 TASK DEFINITIONS

There are five primary ACE recognition tasks – the recognition of ***entities***, ***values***, ***temporal expressions***, ***relations***, and ***events***. These tasks require systems to process language data in documents and then to output, for each of these documents, information about the entities, values, temporal expressions, relations, and events mentioned or discussed in them. This section provides an overview of the ACE tasks. For a complete description refer to the ACE annotation guidelines.[1] The form of the output that is required is defined by an XML format call "APF". The XML DTD for this format may be obtained from the NIST ACE web site.[2]

In addition to the five primary ACE recognition tasks, this year's ACE evaluation will support three mention-level tasks, namely, *entity mentions*, *relation mentions*, and *event mentions*.

2.1 ENTITY DETECTION AND RECOGNITION

The ACE Entity Detection and Recognition task (EDR) requires that certain specified types of entities that are mentioned in the source language data be detected and that selected information about these entities be recognized and merged into a unified representation for each detected entity. The EDR task will be supported for all three ACE languages, which are ***A***rabic, ***C***hinese and ***E***nglish.

2.1.1 ENTITIES

Entity output is required for each document in which the entity is mentioned. This output includes information about the attributes and mentions of the entity. Entity attributes are currently limited to the entity *type*, the entity *subtype*, the entity *class*, and the *name*(s) used to refer to the entity.

The allowable ACE entity types, subtypes and classes for 2005 are listed in Table 1 and Table 2. Entities may have only one type, one subtype and one class. Entity types, subtypes and classes are described in detail in the annotation guidelines. Of the classes discussed in the guidelines, only SPC (specific) entities are assigned a non-zero value during evaluation and therefore systems need output only SPC entities. However, performance on SPC entities may prove to be better if a system attempts to output more than just the SPC entities.

It often happens that different entities may be referred to by the same name. Despite this metonymic connection, however, such entities are regarded as separate and distinct for the purposes of the ACE program. For example, in the sentence "*Miami is growing rapidly*", Miami is a mention of a GPE entity named "Miami", whereas in the sentence "*Miami defeated Atlanta 28 to 3*", Miami is a metonymic mention of an organization entity named "Dolphins" and is distinct from the Miami GPE entity.

Table 1 ACE05 Entity Types and Subtypes

Type	Subtypes
FAC (Facility)	Airport, Building-Grounds, Path, Plant, Subarea-Facility
GPE (Geo-Political Entity[3])	Continent, County-or-District, GPE-Cluster, Nation, Population-Center, Special, State-or-Province
LOC (Location)	Address, Boundary, Celestial, Land-Region-Natural, Region-General, Region-International, Water-Body
ORG (Organization)	Commercial, Educational, Entertainment, Government, Media, Medical-Science, Non-Governmental, Religious, Sports
PER (Person)	Group, Indeterminate, Individual
VEH (Vehicle)	Air, Land, Subarea-Vehicle, Underspecified, Water
WEA (Weapon)	Biological, Blunt, Chemical, Exploding, Nuclear, Projectile, Sharp, Shooting, Underspecified

[1] http://www.ldc.upenn.edu/Projects/ACE/Annotation/

[2] http://www.nist.gov/speech/tests/ace/ace05/doc/

[3] Geo-Political Entities deserve a little supplemental explanation and historical background. Originally, GPE's were not part of the ACE entity inventory. However, during the initial annotation exercises, it became clear that the same word would often imply different entity types – sometimes *location* (as in "the riots in Miami"), sometimes *organization* (as in "Miami imposed a curfew"), sometimes as *person* (as in "Miami railed against the curfew"). Even more troublesome, co-reference was sometimes observed between different underlying entity types (as in "Miami imposed a curfew because of its riots"). These issues gave rise to the definition of the hybrid Geo-Political entity type. This type can be viewed as somewhat synthetic and ad hoc, but there is also support for its conceptual reality, for example by the use of co-reference in joining different entity types.

Table 2 ACE05 Entity Classes

Type	Description
SPC	A particular, specific and unique real world entity
GEN	A kind or type of entity rather than a specific entity
NEG	A negatively quantified (usually generic) entity
USP	An underspecified entity (e.g., modal/uncertain/…)

There are no limits on the use of inference and world knowledge in detecting and recognizing entities. The determination should represent the system's best judgment of the source's intention (i.e., the intention of the author or speaker).

2.1.2 ENTITY MENTIONS

All mentions of each ACE entity are to be detected and output along with the entity attributes. It is important to output every mention to get full value for each entity. The output for each entity mention includes the mention *type*, the location of its *head* and its *extent*, and optionally the mention *role* and *style* of the mention. Mention *style* is either literal or metonymic. This is currently encoded in the apf file format as an attribute called "metonymy_mention", which is either true (for metonymic style of reference) or false (for literal style of reference). The default style is literal. Mention attributes and their possible values are described in detail in the annotation guidelines. The allowable mention types are listed in Table 3.

Table 3 ACE Mention Types

Type	Description
NAM (Name)	A proper name reference to the entity
NOM (Nominal)	A common noun reference to the entity
PRO (Pronominal)	A pronominal reference to the entity

2.2 VALUE DETECTION AND RECOGNITION

The ACE Value Detection and Recognition task (VAL) requires that certain specified types of values that are mentioned in the source language data be detected and that selected information about these values be recognized and merged into a unified representation for each detected value. The VAL task will be supported for two of the ACE languages (Chinese and English). An ACE value is a quantity that provides additional information and that may also be used, as are entities, as arguments of events. Values are represented similarly to entities and are characterized by their attributes and mentions. The type and subtype attributes of each ACE value for 2005 are listed in Table 4. Value types and subtypes are described in detail in the annotation guidelines.

Table 4 ACE05 Value Types and Subtypes

Type	Subtype
Always annotated when mentioned	
Contact-Info	E-Mail, Phone-Number, URL
Numeric	Money, Percent
Annotated when used as an argument in an Event	
Crime	*none*
Job-Title	*none*
Sentence	*none*

2.3 TIME DETECTION AND RECOGNITION

The ACE Time Expression Recognition and Normalization task (TERN) requires that certain temporal expressions mentioned in the source language data be detected and recognized (in timex2 format) according to the *"TIDES 2005 Standard for the Annotations of Temporal Expressions"* April, 2005[4]. The TERN task will be supported for two of the ACE languages (Chinese and English).

Temporal expressions to be recognized include both absolute expressions and relative expressions. In addition, durations, event-anchored expressions and sets of times are to be recognized. This information is contained in the set of timex2 attributes. The ACE timex2 attributes to be evaluated in 2005 are listed in Table 5.

Table 5 ACE05 timex2 attributes

Attribute	Function
VAL	A normalized time expression
MOD	A normalized time expression modifier
ANCHOR_VAL	A normalized time reference point
ANCHOR_DIR	A normalized time directionality
SET	Designates that VAL is a set of time expressions

Note that this year timex2 elements are being reintroduced as arguments of relations and events. Therefore it is important to recognize them and include them as arguments of relations and events where appropriate.

2.4 RELATION DETECTION AND RECOGNITION

The ACE Relation Detection and Recognition task (RDR) requires that certain specified types of relations that are mentioned in the source language data be detected and that selected information about these relations be recognized and merged into a unified representation for each detected relation. The RDR task will be supported for all three ACE languages.

2.4.1 RELATIONS

An ACE relation is a relation between two ACE entities, which are called the relation arguments. Some relations are symmetric, meaning that the ordering of the two entities does not matter (e.g., "partner"). But for asymmetric relations the order does

[4] See http://timex2.mitre.org for more information regarding definition and annotation of timex2 temporal expressions.

matter (e.g., "subsidiary") and for these relations the entity arguments must be assigned the correct argument role.

Relation output is required for each document in which the relation is mentioned. This output includes information about the attributes of the relation, the relation arguments, and the relation mentions. Relation attributes are the relation *type*, *subtype*, *modality* and *tense*. The ACE relation types and subtypes for 2005 are listed in Table 6. Relations may have only one type and one subtype.

Table 6 ACE05 Relation Types and Subtypes
(Relations marked with an * are symmetric relations.)

Type	Subtype
ART (artifact)	User-Owner-Inventor-Manufacturer
GEN-AFF (Gen-affiliation)	Citizen-Resident-Religion-Ethnicity, Org-Location
METONYMY*	*none*
ORG-AFF (Org-affiliation)	Employment, Founder, Ownership, Student-Alum, Sports-Affiliation, Investor-Shareholder, Membership
PART-WHOLE (part-whole)	Artifact, Geographical, Subsidiary
PER-SOC* (person-social)	Business, Family, Lasting-Personal
PHYS* (physical)	Located, Near

2.4.2 RELATION ARGUMENTS

Relation arguments are identified by a unique ID and a role. The roles of the two entities being related are "Arg-1" and "Arg-2" and the correct assignment of these roles to their respective arguments is important, except for symmetric relations (which are identified in Table 6). There may be only one Arg-1 entity and one Arg-2 entity. In addition to the two principal entity arguments there may be one or more temporal (timex2) arguments, and it is important to include these arguments in the relation in order to receive full value for the relation.

2.4.3 RELATION MENTIONS

A relation mention is a sentence or phrase that expresses the relation. The extent of the relation mention is defined to be the sentence or phrase within which the relation is mentioned. A relation mention must contain mentions of both of the entities being related. Although recognition of relation mentions is not evaluated, it is one of the ways that system output relations are allowed to map to reference relations. Thus correct recognition of relation mentions is potentially helpful in evaluation.

2.5 EVENT DETECTION AND RECOGNITION

The ACE Event Detection and Recognition task (VDR) requires that certain specified types of events that are mentioned in the source language data be detected and that selected information about these events be recognized and merged into a unified representation for each detected event. The VDR task will be supported for two ACE languages (Chinese and English).

2.5.1 EVENTS

An ACE event is an event involving zero or more ACE entities, values and time expressions. Event output is required for each document in which the event is mentioned. This output includes information about the attributes of the event, the event arguments, and the event mentions. Event attributes are the event *type*, *subtype*, *modality*, *polarity*, *genericity* and *tense*. The ACE event types and subtypes for 2005 are listed in Table 7. Events may have only one type and one subtype.

Table 7 ACE05 Event Types and Subtypes

Types	Subtype
Life	Be-Born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-Ownership, Transfer-Money
Business	Start-Org, Merge-Org, Declare-Bankruptcy, End-Org
Conflict	Attack, Demonstrate
Contact	Meet, Phone-Write
Personnel	Start-Position, End-Position, Nominate, Elect
Justice	Arrest-Jail, Release-Parole, Trial-Hearing, Charge-Indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

2.5.2 EVENT ARGUMENTS

Each event argument is identified by a unique ID and a role. Unlike relations, which allow only one argument in the Arg-1 and Arg-2 roles, events allow multiple arguments in the same role.

2.5.3 EVENT MENTIONS

An event mention is a sentence or phrase that mentions an event, and the extent of the event mention is defined to be the whole sentence within which the event is mentioned. Although recognition of event mentions is not evaluated, it is one of the ways that system output events are allowed to map to reference events. Thus correct recognition of event mentions is potentially helpful in evaluation.

2.6 ENTITY MENTION DETECTION

The ACE Entity Mention and Detection (EMD) diagnostic task will be supported for all three ACE languages. Section 2.1.2 describes entity mentions.

2.7 RELATION MENTION DETECTION

The ACE Relation Mention and Detection (RMD) diagnostic task will be supported for all three ACE languages. Section 2.4.3 describes relation mentions.

2.8 EVENT MENTION DETECTION

The ACE Event Mention Detection (VMD) diagnostic task will be supported for two of the ACE languages (Chinese and English). Section 2.5.3 describes event mentions.

3 EVALUATION

Evaluation of ACE system performance will be supported for the five primary tasks in three languages. In addition, there will be three diagnostic tasks supported, where partial information is given to the system under test. The evaluation will include

several types of sources and one processing mode, as listed in Table 8.

Table 8 ACE05 Evaluation Support

2005 Evaluation			
Primary Evaluation Tasks:	Languages		
	Ara	Chi	Eng
Entity Detection and Recognition (EDR)	✓	✓	✓
Value Detection and Recognition (VAL)		✓	✓
Timex2 Detection and Recognition (TERN)		✓	✓
Relation Detection and Recognition (RDR)	✓	✓	✓
Event Detection and Recognition (VDR)		✓	✓
Diagnostic Tasks:			
Entity Mention Detection (EMD)	✓	✓	✓
Relation Mention Detection (RMD)	✓	✓	✓
Event Mention Detection (VMD)		✓	✓
EDR Co-Reference (given correct mentions)	✓	✓	✓
RDR given correct entities, values and timex2s	✓	✓	✓
VDR given correct entities, values and timex2s		✓	✓
Processing Mode:			
Document-Level	✓	✓	✓
Cross-Document			
Database Reconciled			
Sources:			
Newswire	✓	✓	✓
Broadcast News	✓	✓	✓
Broadcast Conversations			✓
Weblogs	✓	✓	✓
Usenet Newsgroups/Discussion Forum			✓
Conversational Telephone Speech			✓

Participation is required on at least one of the primary tasks on at least one of the three languages. For each task/language/mode combination chosen, all source material must be processed by the system being evaluated, including all of the different source types contained in the evaluation data.

Performance on each of the different ACE tasks is measured separately. However, since the arguments of relations and events include ACE entities, values and time expressions, a system's performance on relations and events is strongly affected by the system's underlying performance on these elements.

3.1 EVALUATION METHOD

System performance on each of the several tasks is scored using a model of the application value of system output. This overall value is the sum of the value for each system output entity (or value, time expression, relation or event), accumulated over all system outputs. The value of a system output is computed by comparing its attributes and associated information with the attributes and associated information of the reference that corresponds to it. When system output information differs from that of the reference, value is lost. And when system output is spurious (i.e., there is no corresponding reference), negative value typically results. Perfect system output performance is achieved when the system output matches the reference without error. The overall score of a system is computed as the system output information relative to this perfect output. Detail of the valuation of system output and scoring is given in Appendix A – System Output Value Models.

Historically, it has been found that loss of value is attributable mostly to misses (where a reference has no corresponding system output) and false alarms (where a system output has no corresponding reference). To a lesser extent, value is lost due to errors in determining attributes and other associated information in those cases where the system output actually does have a corresponding reference.

3.2 EVALUATION TASKS

3.2.1 ENTITY DETECTION AND RECOGNITION (EDR)

The EDR task is to detect (infer) ACE-defined entities from mentions of them in the source language and to recognize and output selected entity attributes and information about these entities, including information about their mentions. Among other things, this requires that all of the mentions of an entity be correctly associated with that entity. The Value of a system output entity is defined as the product of two factors that represent how accurately the entity's attributes are recognized and how accurately the entity's mentions are detected:

$$Value_{sys\_entity} = Entity\_Value(sys\_entity) \cdot Mentions\_Value(\{sys\_mentions\})$$

Refer to appendix A for a complete description of the EDR *Value* formula.

3.2.2 VALUE DETECTION AND RECOGNITION (VAL)

The VAL task is to detect (infer) ACE-defined value elements from mentions of them in the source language and to recognize and output selected value attributes and information, including information about their mentions. While value elements are currently annotated only at the mention level, both their representation and evaluation are done with the same level of abstraction as that used for entities, namely that value elements are globally unique and may have multiple mentions in multiple documents. The evaluation and scoring of value elements is therefore similar to that for entities. Refer to appendix A for a complete description of the VAL *Value* formula.

3.2.3 TIMEX2 DETECTION AND RECOGNITION (TERN)

The TERN task is to detect (infer) ACE-defined timex2 elements from mentions of them in the source language and to recognize and output selected timex2 attributes and information, including information about their mentions. While timex2 elements are currently annotated only at the mention level, both their

representation and evaluation are done with the same level of abstraction as that used for entities, namely that timex2 elements are globally unique and may have multiple mentions in multiple documents. The evaluation and scoring of timex2 elements is therefore similar to that for entities. Refer to appendix A for a complete description of the timex2 *Value* formula.

3.2.4 RELATION DETECTION AND RECOGNITION (RDR)

The RDR task is to detect (infer) ACE-defined relations from the source language and to recognize and output selected attributes and information about these relations, including information about their mentions and arguments. A major part of correctly recognizing relations is correctly recognizing the arguments that are related by the relation. Therefore good argument recognition performance is important to achieving good RDR performance. The value of a system output relation is defined as the product of two factors that represent how accurately the relation's attributes are recognized and how accurately the relation's arguments are detected and recognized:

$$Value_{sys\_relation} = Relation\_Value(sys\_relation) \cdot Arguments\_Value(\{sys\_arguments\})$$

Refer to appendix A for a complete description of the RDR *Value* formula.

3.2.5 EVENT DETECTION AND RECOGNITION (VDR)

The VDR task is to detect (infer) ACE-defined events from the source language and to recognize and output selected attributes and information about these events, including information about their mentions and arguments. A major part of correctly recognizing events is correctly recognizing the arguments that participate in the event. Therefore good argument recognition performance is important to achieving good VDR performance. The value of a system output event is defined as the product of two factors that represent how accurately the event's attributes are recognized and how accurately the event's arguments are detected and recognized:

$$Value_{sys\_event} = Event\_Value(sys\_event) \cdot Arguments\_Value(\{sys\_arguments\})$$

Refer to appendix A for a complete description of the VDR *Value* formula.

3.2.6 ENTITY MENTION DETECTION (EMD)

The EMD task is to detect (infer) all mentions of ACE-defined entities in the source language and to recognize and output selected attributes and information about these entity mentions. Unlike EDR, EMD does not require that mentions of an entity be correctly associated with an entity. Nevertheless, co-reference remains an important issue because each entity mention must be a mention of an entity within the set of ACE entities.

The EMD value formula is identical to that for EDR. For EMD, however, each entity mention is promoted to "entity" status, separately from other mentions, and thus becomes an entity with only one mention.

3.2.7 RELATION MENTION DETECTION (RMD)

RMD is a derivative task that supports diagnostic evaluation of relation mentions. In RMD, each relation mention, for both system output and reference relations, is promoted to "relation" status and becomes a separate and independent relation and is then evaluated as in RDR. There are several differences between mapping and scoring for RMD and RDR, however. This stems from an inherent ambiguity in specifying the mentions of relation arguments, because often times there are several possible choices. This ambiguity is handled in the following way:

- System output argument mentions are promoted to separate independent argument elements (including entities, values and times). Reference argument mentions are not promoted and are left unchanged as mentions of larger elements. This allows a system argument mention to map to any of the reference argument mentions.

Two other differences between RMD and RDR scoring provide the desired RMD score characteristics:

- Positive overlap is required between reference and system output "extents", defined as the span of their Arg-1/Arg-2 mention heads.
- Argument values are defined to be 1 if the arguments are mappable, 0 otherwise. (A system argument is "mappable" if it has a non-null score with the corresponding reference argument.)

3.2.8 EVENT MENTION DETECTION (VMD)

VMD is a derivative task that supports diagnostic evaluation of event mentions. In VMD, each event mention, for both system output and reference events, is promoted to "event" status and becomes a separate and independent event and is then evaluated as in VDR. There are several differences between mapping and scoring for VMD and VDR, however. This stems from an inherent ambiguity in specifying the mentions of event arguments, because often times there are several possible choices. This ambiguity is handled in the following way:

- System output argument mentions are promoted to separate independent argument elements (including entities, values and times). Reference argument mentions are not promoted and are left unchanged as mentions of larger elements. This allows a system argument mention to map to any of the reference argument mentions.

Two other differences between VMD and VDR scoring provide the desired VMD score characteristics:

- Positive overlap is required between reference and system output mention extents.
- Argument values are defined to be 1 if the arguments are mappable, 0 otherwise. (A system argument is "mappable" if it has a non-null score with the corresponding reference argument.)

3.3 CORPUS SUPPORT

Source language data is being provided to support research (with training corpora that may be subdivided to include a development test set) and evaluation (with an evaluation test corpus). ACE corpora are assembled from a variety of sources selected from broadcast news programs, newspapers, newswire reports, internet sources, and from transcribed audio.

3.3.1 THE ACE 2005 TRAINING CORPUS

The Linguistic Data Consortium has newly annotated ACE training data available[5] for system development. The data is taken from a variety of sources and is available for tasks in all three ACE languages: Arabic, Chinese and English.

ACE05 training and evaluation data was selected using a careful targeted process. Rather than choosing files at random for annotation, as was done in past ACE evaluations, this year's task required a certain density of annotation across the corpus

ACE training corpus statistics including publishing dates are listed in Table 9.

Table 9 2005 ACE system training corpus statistics for release LDC2005E18. This will be an incremental release. Numbers shown represents total size of final release.

Source	Training epoch	Approximate size
English Resources		
Broadcast News	3/03 – 6/03	60,000 words
Broadcast Conversations	3/03 – 6/03	45,000 words
Newswire	3/03 – 6/03	60,000 words
Weblog	11/04 – 2/05	45,000 words
Usenet	11/04 – 2/05	45,000 words
Conversational Telephone Speech	11/04-12/04 (differentiated by topic vs. eval)	45,000 words
Arabic Resources		
Broadcast News	10/00 – 12/00	60,000 words
Newswire	10/00 – 12/00	60,000 words
Weblog	11/04 – 2/05	30,000 words
Chinese Resources *(1.5 characters = 1 word)*		
Broadcast News	10/00 – 12/00	120,000 words
Newswire	10/00 – 12/00	120,000 words
Weblog	11/04 – 2/05	60,000 words

Four versions of each document are provided:

- Source text files (.sgm): All source files, including the Chinese files, are encoded in UTF-8. These files use the UNIX-style end of lines. Only text between the begin text tag <TEXT> and end text tag </TEXT> are to be evaluated. The one exception to this rule is that one TIMEX2 annotation is placed between the <DATETIME> and </DATETIME> tags even though they occur outside the TEXT tags.
- APF files (.apf.xml): The ACE Program Format[6].
- AG files (.ag.xml): The LDC Annotation Graph Format. LDC's internal annotation files format for ACE. These files can be viewed with LDC's annotation tool.
- TABLE files (.tab): Files that store mapping tables between the IDs used in each ag.xml file and their corresponding apf.xml file.

To verify data format integrity, three DTD's are distributed with the ACE training corpus. One DTD is used to verify the APF format, one to verify the AG format, and one to verify the original source document format.

3.3.2 THE 2005 EVALUATION CORPUS

A new evaluation data set is defined for the 2005 evaluation. Table 10 lists the statistics, including the publication dates, of the ACE05 evaluation corpus.

Table 10 The ACE05 evaluation corpus statistics.

Source	Test epoch	Approximate size
English Resources		
Broadcast News	7/03 – 8/03	10,000 words
Broadcast Conversations	7/03 – 8/03	7,500 words
Newswire	7/03 – 8/03	10,000 words
Weblog	3/05 – 4/05	7,500 words
Usenet	3/05 – 4/05	7,500 words
Conversational Telephone Speech	11/04 – 12/04 (different topics from training)	7,500 words
Arabic Resources		
Broadcast News	1/01	20,000 words
Newswire	1/01	20,000 words
Weblog	3/05 – 4/05	10,000 words
Chinese Resources *(1.5 characters = 1 word)*		
Broadcast News	1/01	20,000 words
Newswire	1/01	20,000 words
Weblog	3/05 – 4/05	10,000 words

A key part of system output is the specification of entity mentions in terms of word locations in the source text. Word/phrase location information is in terms of the indices of the first and last characters of the word/phrase. ACE systems must compute these indices from the source data. Indices start with index 0 being assigned to the first character of a document. Ancillary information and annotation, which is provided as bracketed SGML tags, is not included in this count. Only characters (including white-spaces) outside of angle-bracketed expressions contribute to the character count. Also, each new line (nl or cr/lf) counts as one character.

3.3.3 2005 EVALUATION AND SCORING CONDITIONS

All scoring will be done at the document level. This means that each ACE target (entity, time expression, relation, event or value) will contribute to the score for each document that mentions that

[5] Registered participants will be contacted by the LDC with instructions on how to obtain the ACE 2005 training corpus (LDC2005E18).

[6] The ACE APF format is defined by the DTD located at: http://www.nist.gov/speech/tests/ace/ace05/doc/

target. For example, if an entity is mentioned in N different documents, that entity will contribute to the score N times.

All ACE05 tasks will be scored using "document-level processing" mode.

Document-level processing. For this processing mode, each document is processed independently of other documents. No reconciliation ACE targets are required (or allowed), either across documents or with respect to a database. Thus all entities and relations mentioned in a single document must be uniquely associated and identified with that document. This means, by way of example, that if a specific person, say the US president George W. Bush, is mentioned in more than one document, then he must be represented by multiple entities – a different entity (with a globally unique ID) for each document in which he is mentioned.

There are different source conditions depending on the language of the task. Scores will be reported over the entire evaluation test set as well as separately for each source domain. This will support contrasts between different sources.

3.4 RULES

- No changes to the system are allowed once the evaluation data are released. Adaptive systems may of course change themselves in response to the source data that they process.
- No human intervention is allowed prior to the submission of your test site's results to NIST.[7] This means that, in addition to disallowing modifications to your system, there must also be no modifications to, or human examination of, the test data.
- For each evaluation combination of task, language, and processing mode for which system output is submitted, all documents from all sources for that evaluation combination must be processed.
- NIST will email the evaluation test data to each site on 11/09/05. Sites must return results to NIST within a 24 hour period. The actual starting time on 11/09 is negotiable[8].
- Every participating site must submit a detailed system description to NIST by 11/30/05, as defined in section 3.7.2.
- Every participating site must attend the evaluation workshop and present a system talk if requested[9].

[7] It sometimes happens that a system bug is discovered during the course of processing the test data. In such a case, please consult with NIST email (ace_poc@nist.gov) for advice. NIST will advise you on how to proceed. Repairs may be possible that allow a more accurate assessment of the underlying performance of a system. If this happens, modified results may be accepted, provided that an explanation of the modification is provided and provided that the original results are also submitted and documented.

[8] By default, NIST will send the evaluation data to the registered participants at 9:00am EST, with results due back 24 hours later. It may be desirable for some sites to receive the data at some other time on 11/09/05. It is the registered sites responsibility to contact NIST (ace_poc@nist.gov) to schedule the exact time of data delivery.

[9] Note, not all participants will be requested to give a site talk. The workshop will include a poster session where everyone will have the opportunity to discuss their work. The number of site talks will depend on the number of participants, the innovations of their system algorithms, the tasks and languages attempted, and the quality of the results.

3.5 TOOLS

3.5.1 XML VALIDATION TOOLS

A java implementation of an XML validator[10] is available from the NIST ACE web site. The XML validator will verify that a system output file conforms to the current ACE DTD.[11]

Before sites submit their system results to NIST for scoring, they must validate the results file using the XML validation tool and the current ACE APF DTD. ***Results that are not validated will not be accepted.***

3.5.2 ACE EVALUATION SOFTWARE

The ACE evaluation software is available for download from the NIST ACE web site.[12] This tool scores EDR, VAL, TERN, RDR, and VDR output.

3.6 SCHEDULE

Table 11 The ACE 2005 Evaluation Schedule

Date	Event
11/01/05	Deadline to register[13] for participation in the ACE05 evaluation.
11/07/05	ACE05 Arabic evaluation day
11/08/05	ACE05 Chinese evaluation day
11/09/05	ACE05 English evaluation day
11/14/05	Ground-truth entity mentions available for diagnostic EDR task
11/16/05	(noon deadline, EST) Diagnostic EDR results due at NIST
11/16/05	Ground-truth ENTITIES available for diagnostic RDR and VDR tasks
11/18/05	(COB deadline) Diagnostic RDR and VDR results due at NIST
11/23/05	NIST releases results
11/30/05	(noon deadline, EST) Site's detailed system description papers are due at NIST
12/05/05	A handful of sites will receive requests to give formal talks at the evaluation workshop.
12/15-16/05	Two day evaluation workshop.

[10] URL: http://www.nist.gov/speech/tests/ace/ace05/software.htm

[11] The DTD's used for the ACE program, can be found at: http://www.nist.gov/speech/tests/ace/dtd/

[12] The ACE evaluation tools may be accessed from the NIST ACE URL http://www.nist.gov/speech/tests/ace/ace05/software.htm

[13] The official ACE05 registration form is located at the URL: http://www.nist.gov/speech/tests/ace/ace05/doc/

3.7 SUBMISSION OF SYSTEM OUTPUT TO NIST

To enable quick unpacking and scoring of several site submission files with minimum human intervention, participants must follow the outlined procedure for submitting results.

3.7.1 PACKAGING YOUR SYSTEM OUTPUT

Note, that in many cases a system output file will contain results for more than one task (i.e. EDR and RDR). In such a case the exact same set of files should be copied to the EDR and RDR subdirectories as defined below.

STEP1: Create a top level directory for each of the ***languages*** attempted (Arabic | Chinese | English):

Example: $> mkdir chinese english

STEP2: Create a subdirectory identifying the ***tasks*** attempted (EDR | VAL | TERN | RDR| VDR):

Example: $> mkdir english/edr english/rdr chinese/edr

STEP3: In each of these subdirectories make one directory for each system submitted (choose a name that identifies your site, BBN, SHEF, SRI…):

Example: $> mkdir english/edr/NIST1_primary

Example: $> mkdir english/edr/NIST2_contrastive1

Example: $> mkdir english/rdr/NIST1_primary

Example: $> mkdir chinese/edr/NIST1_primary

STEP4: Deposit all system output files in the appropriate system directory.

STEP5: Create a compressed tar file of your results and transfer them to NIST by FTP (ftp://ijaguar.ncsl.nist.gov/incoming). After successful transmission send e-mail to ace_poc@nist.gov identifying the name of the file submitted. Alternatively you may send the compressed tar file directly to ace_poc@nist.gov .

3.7.2 SYSTEM DESCRIPTION

A valuable tool in discovering strengths and weakness of different algorithmic approaches is the use of system descriptions. This year, system descriptions will also be used to help determine which sites are to give oral workshop presentations and which sites are to give talks in a poster session.

Each participant must prepare a ***detailed*** system description covering each system submitted. System descriptions are due at NIST no later than 11/30/05. It is important that all sites submit comprehensive descriptions on time so that NIST may plan the workshop agenda accordingly.

These system descriptions will be distributed to each participant before the evaluation workshop.

Each system description should include:

- The ACE tasks and languages processed
- Identification of the primary system for each task
- A description of the system (algorithms, data, configuration) used to produce the system output
- How contrastive systems differ from the primary system
- A description of the resources required to process the test set, including CPU time and memory
- Applicable references

4 GUIDELINES FOR PUBLICATIONS

NIST Speech Group's HLT evaluations are moving towards an open model which promotes interchange with the outside world. Therefore, the rules governing the publication of ACE05 evaluation results have been updated..

4.1 NIST PUBLICATION OF RESULTS

At the conclusion of the evaluation cycle, NIST will create a report which documents the evaluation. The report will be posted on the NIST web space and will identify the participants and official ACE value scores achieved for each task/language combination. Scores will be reported for the overall test set and for the different data sources.

The report that NIST creates should not be construed, or represented as endorsements for any participant's system or commercial product, or as official findings on the part of NIST or the U.S. Government.

4.2 PARTICIPANT'S PUBLICATION OF RESULTS

Participants will be free to publish results for their own system, and may state the highest score achieved in a particular task, but sites will not be allowed to name other participants, or cite another site's results without permission from the other site. Publications may point to the NIST report as a reference[14].

[14] This restriction exists to ensure that readers concerned with a particular system's performance will see the entire set of participants and tasks attempted by all researchers.

APPENDIX A – SYSTEM OUTPUT VALUE MODELS

EDR SCORING

The EDR value score for a system is defined to be the sum of the values of all of the system's output entity tokens, normalized by the sum of the values of all reference entity tokens. The maximum possible EDR value score is 100 percent.

$$EDR\_Value_{sys} = \sum_i value\_of\_sys\_token_i \Big/ \sum_j value\_of\_ref\_token_j$$

The value of each system token is based on its attributes and on how well it matches its corresponding reference token. A globally optimum correspondence between system and reference tokens which maximizes *EDR_Value* is determined and used, subject to the constraint of one-to-one mapping between system and reference tokens.[15] The value of a system token is defined as the product of two factors that represent both the inherent value of the token and how accurately the token's attributes are recognized and the token's mentions are detected.

$$Value(token) = Element\_Value(token) \cdot Mentions\_Value(token)$$

Element_Value is a function of the attributes of the system token and, if mapped, how well they match those of the corresponding reference token. The inherent value of a token is defined as the product of the token's attribute value parameters, *AttrValue*, for the attributes ***type*** and ***class***. This inherent value is reduced for any attribute errors (i.e., for any differences between the values of system and reference attributes), using the error weighting parameters, $\{W_{err\text{-}attribute}\}$. If a system token is unmapped, then the value of that token is weighted by a false alarm penalty, $W_{E\text{-}FA}$.

$$Element\_Value(sys) = \begin{cases} \min\left(\prod_{\substack{attribute= \\ type,class}} AttrValue(attribute_{sys}), \prod_{\substack{attribute= \\ type,class}} AttrValue(attribute_{ref}) \right) \cdot \prod_{\substack{attribute= \\ type,subtype,class}} W_{err-attribute} & \text{if } sys \text{ mapped} \\ \left(\prod_{\substack{attribute= \\ type,class}} AttrValue(attribute_{sys}) \right) \cdot W_{FA} & \text{if not mapped} \end{cases}$$

Mentions_Value is a function of the mutual mention value (*MMV*) between the mentions of the system token and, if mapped, those of the corresponding reference token[16]. A mention's *MMV* depends on the mention's type value parameter, *MTypeValue*, with this value being reduced for any mention attribute errors (i.e., for any differences between the attribute values of system and reference mentions), W_{Merr}. If the system mention is unmapped, then the *MMV* is weighted by a false alarm penalty factor, $W_{M\text{-}FA}$, and also by a co-reference weighting factor, $W_{M\text{-}CR}$, if the system mention happens to correspond to a legitimate reference mention but one that doesn't belong to the corresponding reference token[17].

$$MMV(mention_{sys}) = \begin{cases} \min\left(MTypeValue(mention_{sys}), MTypeValue(mention_{ref}) \right) \cdot \prod_{\substack{attribute= \\ type,role,style}} W_{Merr-attribute} & \text{if } mention_{sys} \text{ mapped} \\ -MTypeValue(mention_{sys}) \cdot (W_{M-FA} \cdot W_{M-CR}) & \text{if not mapped} \end{cases}$$

For each pairing of a system token with a reference token, an optimum correspondence between system mentions and reference mentions that maximizes the sum of *MMV* over all system mentions is determined and used, subject to the constraint of one-to-one mapping between system and reference mentions.

Mentions_Value is computed using one of two formulas, depending on whether valuation is ***mention***-weighted or ***level***-weighted. For mention-weighted valuation *Mentions_Value* is simply the sum of *MMV* over all mentions in all documents. For level-weighted valuation *Mentions_Value* is determined by a system token's "level" (and that of its corresponding reference token) and by the number of documents in which the token is mentioned. The "level" of a token is the highest (i.e., the most valued) mention type of that token. Thus, for example, the "level" of a token is NAM (named) if any one of its mentions is of type NAM, because NAM mentions are more

---

[15] System tokens and reference tokens are permitted to correspond only if they each have at least one mention in correspondence with the other.

[16] All mentions of a system token are considered to be unmapped for tokens that are themselves unmapped. Thus, for tokens that are unmapped, *Mentions_Value* will be negative. (Note the minus sign in the formula for the *MMV* of unmapped mentions.)

[17] This is intended to avoid double penalizing co-reference errors, namely once for missing the mention in the correct token and once for including the mention in the wrong token. Setting $W_{M\text{-}CR}$ to zero eliminates the second penalty.

valuable than NOM mentions. If none of its mentions is of type NAM, but at least one mention is of type NOM, then the "level" of that token would be NOM (nominal).

$$Mentions\_Value(sys) = \begin{cases} \sum_{\substack{all\\docs}} \left(\sum_{\substack{all\ sys\\mentions\\in\ doc}} MMV(m_{sys}) \right) & \text{if mention - weighted} \\ \min\left(\begin{matrix} MTypeValue(level_{sys}) \\ MTypeValue(level_{ref}) \end{matrix} \right) \cdot \sum_{\substack{all\\docs}} \left(\sum_{\substack{all\ sys\\mentions\\in\ doc}} MMV(m_{sys}) \Big/ \sum_{\substack{all\ ref\\mentions\\in\ doc}} MMV(m_{ref}) \right) & \text{if level - weighted} \end{cases}$$

System mentions and reference mentions are permitted to correspond only if their **heads** have a mutual overlap of at least *min_overlap* and the text of their **heads** share a (fractional) consecutive string of characters[18] of at least *min_text_match*. Mention regions and overlaps are measured in terms of *number of characters* for text input, in terms of *time* for audio input, and in terms of *area* for image input.

$$mutual\_overlap = \frac{sys\_head \cap ref\_head}{\max(sys\_head, ref\_head)}$$

$$fractional\_consecutive\_string = \frac{\left(\begin{matrix} \#\ of\ characters\ in\ the\ longest\ consecutive\ string\ of\ characters \\ that\ is\ contained\ in\ both\ system\ and\ reference\ mention\ head\ texts \end{matrix} \right)}{\max\left(\begin{matrix} \#\ of\ characters\ in\ system\ mention\ head\ text, \\ \#\ of\ characters\ in\ reference\ mention\ head\ text \end{matrix} \right)}$$

The current default scoring parameters for EDR are given in Table 12.

Table 12 Default parameters for scoring EDR performance

***Element_Value* parameters**			
Attribute	$W_{err\text{-}attribute}$	**Attribute Value**	***AttrValue***
Type	0.50	(all types)	1.00
Class	0.75	SPC	1.00
		(not SPC)	0.00
Subtype	0.90	n/a	n/a
$W_{E\text{-}FA}$ = 0.75			
***Mentions_Value* parameters**			
Attribute	$W_{Merr\text{-}attribute}$	**Attribute Value**	***MTypeValue***
Type	0.90	NAM	1.00
		NOM	0.50
		PRO	0.10
Role	0.90	n/a	n/a
Style	0.90	n/a	n/a
Valuation = level-weighted			
$W_{M\text{-}FA}$ = 0.75		$W_{M\text{-}CR}$ = 0.00	
min_overlap = 0.30		***min_text_match*** = 0.30	

[18] This requirement of a common substring in both system and output mention heads was invoked to account for errors in transcribing speech and image data into text. The intent is to require a mention be meaningful and relevant in order to be counted.

VAL SCORING

The VAL value score for a system is defined to be the sum of the values of all of the system's output value tokens, normalized by the sum of the values of all of the reference value tokens. The maximum possible VAL value score is 100 percent.

$$VAL\_Value_{sys} = \sum_i value\_of\_sys\_token_i \Big/ \sum_j value\_of\_ref\_token_j$$

The value of each system token is based on its attributes and on how well it matches its corresponding reference token. A globally optimum correspondence between system and reference tokens which maximizes *VAL_Value* is determined and used, subject to the constraint of one-to-one mapping between system and reference tokens.[15] The value of a system token is defined as the product of two factors that represent both the inherent value of the token and how accurately the token's attributes are recognized and the token's mentions are detected.

$$Value(token) = Element\_Value(token) \cdot Mentions\_Value(token)$$

Element_Value depends on the token type and, if mapped, on how well the attributes of the system token match those of the corresponding reference token. The inherent value of a token is determined by the token's type value parameter, *AttrValue*(*type*). This inherent value is reduced for any attribute errors (i.e., for any differences between the values of system and reference attributes), using the error weighting parameters, $\{W_{err\text{-}attribute}\}$. If a system token is unmapped, then the value of that token is weighted by a false alarm penalty, W_{FA}.

$$Element\_Value(sys) = \begin{cases} \min\begin{pmatrix} AttrValue(type_{sys}), \\ AttrValue(type_{ref}) \end{pmatrix} \cdot \prod_{\substack{attribute= \\ type,subtype}} W_{err-attribute} & \text{if } sys \text{ mapped} \\ \left(AttrValue(type_{sys})\right) \cdot W_{FA} & \text{if not mapped} \end{cases}$$

Mentions_Value is simply the sum of the mutual mention value (*MMV*) between the mentions of the system token and, if mapped, those of the corresponding reference token[16]. A mention's *MMV* is simply the value 1. If the system mention is unmapped, then the *MMV* is weighted by a false alarm penalty factor, $W_{M\text{-}FA}$, and also by a co-reference weighting factor, $W_{M\text{-}CR}$, if the system mention happens to correspond to a legitimate reference mention but one that doesn't belong to the corresponding reference token[17]. For each pairing of a system token and a reference token, an optimum correspondence between system mentions and reference mentions that maximizes the sum of *MMV* over all system mentions is determined and used, subject to the constraint of one-to-one mapping between system and reference mentions.

$$MMV(mention_{sys}) = \begin{cases} 1 & \text{if } mention_{sys} \text{ mapped} \\ -(W_{M-FA} \cdot W_{M-CR}) & \text{if not mapped} \end{cases} \qquad Mentions\_Value(sys) = \sum_{\substack{all \\ docs}} \left(\sum_{\substack{all\ sys \\ mentions \\ in\ doc}} MMV(m_{sys}) \right)$$

System mentions and reference mentions are permitted to correspond only if their **extents** have a mutual overlap of at least *min_overlap*. Mention regions and overlaps are measured in terms of *number of characters* for text input, in terms of *time* for audio input, and in terms of *area* for image input.

$$mutual\_overlap = \frac{sys\_extent \cap ref\_extent}{\max(sys\_extent, ref\_extent)}$$

The current default parameters for VAL scoring are given in Table 13.

Table 13 Default parameters for scoring VAL performance

***Element_Value* parameters**				***Mentions_Value* parameters**	
Attribute	$W_{err\text{-}attribute}$	**Attribute Value**	***AttrValue***	$W_{Merr\text{-}attribute}$	0.90 (for all attributes)
Type	0.50	(all types)	1.00	$W_{M\text{-}FA}$	0.75
Subtype	0.90	n/a	n/a	$W_{M\text{-}CR}$	0.00
W_{FA} = 0.75				***min_overlap***	0.30

TERN SCORING

The TERN value score for a system is defined to be the sum of the values of all of the system's output timex2 tokens, normalized by the sum of the values of all of the reference timex2 tokens. The maximum possible timex2 value score is 100 percent.

$$TERN\_Value_{sys} = \sum_i value\_of\_sys\_token_i \Big/ \sum_j value\_of\_ref\_token_j$$

The value of each system token is based on its attributes and on how well it matches its corresponding reference token. A globally optimum correspondence between system and reference tokens which maximizes *TERN_Value* is determined and used, subject to the constraint of one-to-one mapping between system and reference tokens.[15] The value of a system token is defined as the product of two factors that represent both the inherent value of the token and how accurately the token's attributes are recognized and the token's mentions are detected.

$$Value(token) = Element\_Value(token) \cdot Mentions\_Value(token)$$

Element_Value depends on how well the attributes of the system token match those of the corresponding reference token. The inherent value of a token is defined as a sum of attribute value parameters, *AttrValue*, summed over all attributes which exist and which are the same for both the system and reference tokens. If a system token is unmapped, then the value of that token is weighted by a false alarm penalty, W_{FA}.

$$Element\_Value(sys) = \begin{cases} \sum_{\substack{\text{for all existing } sys \text{ attributes in the set} \\ \{type, mod, set, val, anchor\_dir, anchor\_val\}}} \begin{cases} AttrValue(attribute) & \text{if } attribute_{sys} = attribute_{ref} \\ 0 & \text{otherwise} \end{cases} & \text{if } sys \text{ mapped} \\ \sum_{\substack{\text{for all existing } sys \text{ attributes in the set} \\ \{type, mod, set, val, anchor\_dir, anchor\_val\}}} AttrValue(attribute) \cdot W_{FA} & \text{if not mapped} \end{cases}$$

Mentions_Value is simply the sum of the mutual mention value (*MMV*) between the mentions of the system token and, if mapped, those of the corresponding reference token[16]. A mention's *MMV* is simply the value 1. If the system mention is unmapped, then the *MMV* is weighted by a false alarm penalty factor, $W_{M\text{-}FA}$, and also by a co-reference weighting factor, $W_{M\text{-}CR}$, if the system mention happens to correspond to a legitimate reference mention but one that doesn't belong to the corresponding reference token[17]. For each pairing of a system token and a reference token, an optimum correspondence between system mentions and reference mentions that maximizes the sum of *MMV* over all system mentions is determined and used, subject to the constraint of one-to-one mapping between system and reference mentions.

$$MMV(mention_{sys}) = \begin{cases} 1 & \text{if } mention_{sys} \text{ mapped} \\ -(W_{M-FA} \cdot W_{M-CR}) & \text{if not mapped} \end{cases} \qquad Mentions\_Value(sys) = \sum_{\substack{all \\ docs}} \left(\sum_{\substack{all\ sys \\ mentions \\ in\ doc}} MMV(m_{sys}) \right)$$

System mentions and reference mentions are permitted to correspond only if their **extents** have a mutual overlap of at least *min_overlap*. Mention regions and overlaps are measured in terms of *number of characters* for text input, in terms of *time* for audio input, and in terms of *area* for image input.

$$mutual\_overlap = \frac{sys\_extent \cap ref\_extent}{\max(sys\_extent, ref\_extent)}$$

The current default parameters for TERN scoring are given in Table 14.

Table 14 Default parameters for scoring TERN performance

***Element_Value* parameters**						
attribute	type	anchor_dir	anchor_val	mod	set	val
AttrValue	0.10	0.25	0.50	0.10	0.10	1.00
$W_{E\text{-}FA} = 0.75$						
***Mentions_Value* parameters**						
$W_{M\text{-}FA} = 0.75$		$W_{M\text{-}CR} = 0.00$			***min_overlap*** = 0.30	

RDR SCORING

The RDR value score for a system is defined to be the sum of the values of all of the system's output relation tokens, normalized by the sum of the values of all reference relation tokens. The maximum possible RDR value score is 100 percent.

$$RDR\_Value_{sys} \;=\; \sum_i value\_of\_sys\_token_i \;\Big/\; \sum_j value\_of\_ref\_token_j$$

The value of each system token is based on its attributes and arguments and on how well they match those of a corresponding reference token. A globally optimum correspondence between system and reference tokens which maximizes *RDR_Value* is determined and used, subject to the constraint of one-to-one mapping between system and reference tokens. System tokens and reference tokens are permitted to correspond only if they have some nominal basis for correspondence. The required nominal basis is selectable from the set of minimal conditions listed in Table 15.

Table 15 Conditions required for correspondence between system and reference relation tokens

Condition	Description
arguments	At least one argument in the system token must be mappable to an argument in the reference token.
extents	The system and reference tokens must each have at least one mention extent in correspondence with the other.
both	Both the **arguments** condition and the **extents** condition must be met.
either	Either the **arguments** condition or the **extents** condition must be met.
all	All arguments in the reference token must be one-to-one mappable to arguments in the system token.
all+extents	Both the **all** condition and the **extents** condition must be met.

The value of a system token is defined as the product of two factors that represent both the inherent value of the token and how accurately the token's attributes and arguments are recognized.

$$Value(token) \;=\; Element\_Value(token) \;\cdot\; Arguments\_Value(token)$$

Element_Value is a function of the attributes of the system token and, if mapped, how well they match those of the corresponding reference token. The inherent value of a token is defined as the product of the token's attribute value parameters, *AttrValue*, for the attributes ***type*** and ***modality***. This inherent value is reduced for any attribute errors (i.e., for any difference between the values of system and reference attributes), using the error weighting parameters, $\{W_{err\text{-}attribute}\}$. If a system token is unmapped, then the value of that token is weighted by a false alarm penalty, W_{FA}.

$$Element\_Value(sys) \;=\; \begin{cases} \min\left(\prod_{\substack{attribute= \\ type,modality}} AttrValue(attribute_{sys}), \prod_{\substack{attribute= \\ type,modality}} AttrValue(attribute_{ref}) \right) \cdot \prod_{\substack{attribute= \\ type,subtype,modality,tense}} W_{err-attribute} & \text{if mapped} \\ AttrValue(attribute_{sys}) \cdot W_{FA} & \text{if not mapped} \end{cases}$$

Arguments_Value is a function of the mutual argument value (*MAV*) between the arguments of the system token and, if mapped, those of the corresponding reference token.[19] An argument's *MAV* depends on the system argument's value (with respect to the putative reference argument) for each document in which the relation is mentioned, $Value_{doc}(arg_{sys},arg_{ref})$, with this value being reduced for argument role errors (i.e., for a difference between the roles of system and reference arguments), $W_{err\text{-}role}$. Argument-level errors are accounted for using an incremental formulation of false alarm error. Specifically, loss of value at the argument level is viewed as a partial false alarm, and this loss of value is subtracted from the *MAV* after being weighted by a false alarm penalty factor, $W_{A\text{-}FA}$.

$$MAV_{doc}(arg_{sys}) \;=\; Value_{doc}(arg_{sys}, arg_{ref}) \cdot W_{err-role} \;-\; \left(Value_{doc}(arg_{sys}, arg_{sys}) - Value_{doc}(arg_{sys}, arg_{ref})\right) \cdot W_{A-FA}$$

[19] All arguments of a system token are considered to be unmapped for tokens that are themselves unmapped. Thus, for tokens that are unmapped, *Arguments_Value* will be negative. Note that *MAV* is negative for unmapped arguments, i.e., when $Value_{doc}(arg_{sys},arg_{ref}) = 0$.

If there is no corresponding reference argument for a system argument, then $Value_{doc}(arg_{sys},arg_{ref})$ is taken to be zero. There are several requirements that must be satisfied in order for a reference argument to be considered to be in correspondence to a system argument. First, note that there are two required arguments, namely the two arguments for which the relation is being asserted. These arguments have roles called "Arg-1" and "Arg-2", and there may be only one Arg-1 and one Arg-2 argument.[20] The requirements for correspondence are listed in Table 16.

Table 16 Conditions required for correspondence between system and reference relation arguments

Condition	Requirement
Always	The reference argument must be mappable to the system argument. That is, they must have at least one mention in correspondence.
If the "**mapped**" argument option is invoked	The reference argument must correspond to the system argument. That is, they must be mapped to each other at the argument level.
Argument role is Arg-1 or Arg-2 and the relation is **not** "symmetric"	The reference argument role must be the same as the system argument role.
Argument role is Arg-1 or Arg-2 and the relation **is** "symmetric"[21]	The reference argument role must be either "Arg-1" or "Arg-2".

For each pairing of a system relation token with a reference relation token, an optimum correspondence between system arguments and reference arguments that maximizes *Arguments_Value* is determined and used. This optimum mapping is constrained to be a one-to-one mapping between system and reference arguments.

Arguments_Value is computed using one of two formulas, depending on whether the contribution of the various relation arguments are averaged arithmetically or geometrically.

$$Arguments\_Value(sys) = \left\{ \begin{array}{l} \sum_{all\ arg_{sys}} \left(\sum_{\substack{all\ docs\ that \\ mention\ the\ relation}} MAV_{doc}\left(arg_{sys},arg_{ref}\right) \right) \quad \text{if arithmetic averaging} \\ \prod_{all\ arg_{sys}} \left(\sum_{\substack{all\ docs\ that \\ mention\ the\ relation}} MAV_{doc}\left(arg_{sys},arg_{ref}\right) \right) \quad \text{if geometric averaging} \end{array} \right\}$$

Note that geometric averaging is sensible only when the *MAV* value contributions exist and are positive for all reference arguments. Thus, for geometric averaging, all reference arguments must be mapped (condition **all** or **all+extents** in Table 15) and $W_{A\text{-}FA}$ must be 0.

The current default scoring parameters for RDR are given in Table 17.

Table 17 Default parameters for scoring RDR performance

***Element_Value* parameters**				
Attribute	Type	Subtype	Modality	Tense
AttrValue	1.00 for all types	n/a	1.00 for all modalities	n/a
$W_{err\text{-}attribute}$	1.00	0.70	0.75	1.00
***Relation mapping requirements* (**Table 15**) = "arguments"**				
W_{FA} = 0.75				
***Arguments_Value* parameters**				
"mapped" *arguments optional requirement* NOT *invoked* (Table 16**)**				
Both Arg-1 and Arg-2 arguments must be mappable (i.e., must have non-null *MAV*'s)				
"arithmetic" *averaging of argument scores*				
$W_{err\text{-}role}$ = 0.75		$W_{A\text{-}FA}$ = 0.00		

[20] Arg-1 and Arg-2 are the only roles for which the number of arguments is limited.

[21] For the 2005 evaluation, the only symmetric relations are those of type "PER-SOC", "PHYS", and "METONMY".

VDR SCORING

The VDR value score for a system is defined to be the sum of the values of all of the system's output event tokens, normalized by the sum of the values of all reference event tokens. The maximum possible VDR value score is 100 percent.

$$VDR\_Value_{sys} = \sum_i value\_of\_sys\_token_i \Big/ \sum_j value\_of\_ref\_token_j$$

The value of each system token is based on its attributes and arguments and on how well they match those of a corresponding reference token. A globally optimum correspondence between system and reference tokens which maximizes *VDR_Value* is determined and used, subject to the constraint of one-to-one mapping between system and reference tokens. System tokens and reference tokens are permitted to correspond only if they have some nominal basis for correspondence. The required nominal basis is selectable from the set of minimal conditions listed in Table 18. Note that the condition selected applies to both VDR and RDR.

Table 18 Conditions required for correspondence between system and reference event tokens

Condition	Description
arguments	At least one argument in the system token must be mappable to an argument in the reference token.
extents	The system and reference tokens must each have at least one mention extent in correspondence with the other.
both	Both the **arguments** condition and the **extents** condition must be met.
either	Either the **arguments** condition or the **extents** condition must be met.
all	All arguments in the reference token must be one-to-one mappable to arguments in the system token.
all+extents	Both the **all** condition and the **extents** condition must be met.

The value of a system token is defined as the product of two factors that represent both the inherent value of the token and how accurately the token's attributes and arguments are recognized.

$$Value(token) = Element\_Value(token) \cdot Arguments\_Value(token)$$

Element_Value is a function of the attributes of the system token and, if mapped, how well they match those of the corresponding reference token. The inherent value of a token is defined as the product of the token's attribute value parameters, *AttrValue*, for the attributes ***type*** and ***modality***. This inherent value is reduced for any attribute errors (i.e., for any difference between the values of system and reference attributes), using the error weighting parameters, $\{W_{err\text{-}attribute}\}$. If a system token is unmapped, then the value of that token is weighted by a false alarm penalty, W_{FA}.

$$Element\_Value(sys) = \begin{cases} \min\left(\prod_{\substack{attribute= \\ type,modality}} AttrValue(attribute_{sys}), \prod_{\substack{attribute= \\ type,modality}} AttrValue(attribute_{ref}) \right) \cdot \prod_{\substack{attribute= \\ type,subtype,modality, \\ genericity,polarity,tense}} W_{err-attribute} & \text{if mapped} \\ AttrValue(attribute_{sys}) \cdot W_{FA} & \text{if not mapped} \end{cases}$$

Arguments_Value is a function of the mutual argument value (*MAV*) between the arguments of the system token and, if mapped, those of the corresponding reference token.[22] An argument's *MAV* depends on the system argument's value (with respect to the putative reference argument) for each document in which the event is mentioned, $Value_{doc}(arg_{sys},arg_{ref})$, with this value being reduced for argument role errors (i.e., for a difference between the roles of system and reference arguments), $W_{err\text{-}role}$. Argument-level errors are accounted for using an incremental formulation of false alarm error. Specifically, loss of value at the argument level is viewed as a partial false alarm, and this loss of value is subtracted from the *MAV* after being weighted by a false alarm penalty factor, $W_{A\text{-}FA}$.

$$MAV_{doc}(arg_{sys}) = Value_{doc}(arg_{sys}, arg_{ref}) \cdot W_{err-role} - \left(Value_{doc}(arg_{sys}, arg_{sys}) - Value_{doc}(arg_{sys}, arg_{ref})\right) \cdot W_{A-FA}$$

[22] All arguments of a system token are considered to be unmapped for tokens that are themselves unmapped. Thus, for tokens that are unmapped, *Arguments_Value* will be negative. Note that *MAV* is negative for unmapped arguments, i.e., when $Value_{doc}(arg_{sys},arg_{ref}) = 0$.

If there is no corresponding reference argument for a system argument, then $Value_{doc}(arg_{sys},arg_{ref})$ is taken to be zero. There are several requirements that must be satisfied in order for a reference argument to be considered to be in correspondence to a system argument. These requirements for correspondence are listed in Table 19.

Table 19 Conditions required for correspondence between system and reference event arguments

Condition	Requirement
Always	The reference argument must be mappable to the system argument. That is, they must have at least one mention in correspondence.
If the "**mapped**" argument option is invoked	The reference argument must correspond to the system argument. That is, they must be mapped to each other at the argument level.

For each pairing of a system event token with a reference event token, an optimum correspondence between system arguments and reference arguments that maximizes *Arguments_Value* is determined and used. This optimum mapping is constrained to be a one-to-one mapping between system and reference arguments.

Arguments_Value is computed using one of two formulas, depending on whether the contribution of the various event arguments are averaged arithmetically or geometrically.

$$Arguments\_Value(sys) = \left\{ \begin{array}{l} \sum_{all\ arg_{sys}} \left(\sum_{\substack{all\ docs\ that \\ mention\ the\ relation}} MAV_{doc}(arg_{sys},arg_{ref}) \right) \quad \text{if arithmetic averaging} \\ \prod_{all\ arg_{sys}} \left(\sum_{\substack{all\ docs\ that \\ mention\ the\ relation}} MAV_{doc}(arg_{sys},arg_{ref}) \right) \quad \text{if geometric averaging} \end{array} \right\}$$

Note that geometric averaging is sensible only when the *MAV* value contributions exist and are positive for all reference arguments. Therefore, for geometric averaging, all reference arguments must be mapped (condition **all** or **all+extents** in Table 18) and $W_{A\text{-}FA}$ must be zero.

The current default scoring parameters for VDR are given in

Table 20 Default parameters for scoring VDR performance

***Element_Value* parameters**						
Attribute	Type	Subtype	Modality	Genericity	Polarity	Tense
AttrValue	1.00 for all types	n/a	1.00 for all modalities	n/a	n/a	n/a
$W_{err\text{-}attribute}$	0.50	0.90	0.75	1.00	1.00	1.00
***Event mapping requirements* (**Table 18**) = "arguments"**						
W_{FA} = 0.75						
***Arguments_Value* parameters**						
"mapped" *arguments optional requirement* NOT *invoked* (Table 19**)**						
"arithmetic" *averaging of argument scores*						
$W_{err\text{-}role}$ = 0.75			$W_{A\text{-}FA}$ = 0.50			